Clean copy of Allowed Claims

11. A method in a LSI design and development process for creating and evaluating an architecture design for an algorithm design by performing a performance evaluation of at least one bus at a high-level stage of said design and development process, said method comprising:

structuring source code describing the algorithm design in a general purpose high-level programming language by isolating elements of said source code into elements representing hardware units and elements representing software units creating the architecture design of an LSI;

creating an evaluation function for counting data traffic that occurs on said at least one bus, the bus being a part of the source code realizing the data traffic between said elements representing hardware units and software units;

sequentially reading in the source code line by line while effecting syntax analysis;

determining whether the source code is to be modified based on whether a line of source code represents writing data to variables that are defined in advance and are loaded onto the bus to be evaluated;

modifying at least one element of said source code based on an implementation of said evaluation function; and

performing said performance evaluation by simulating said modified source code elements and counting said data traffic on the bus, wherein said performance evaluation is used to modify the architecture design of the LSI.

12. The method according to claim 11, further comprising:

restructuring the source code based on the counted data traffic; and

performing said performance evaluation again by simulating said restructured source code again.

13. The method according to claim 11, wherein a bus traffic is calculated from the counted data traffic using a processing rate of the bus.

14. The method according to claim 11, further comprising:

feeding back a result of the performance evaluation of the bus to the step of structuring the source code to improve the architecture design at the high-level design stage by isolating elements of said source code into new elements representing hardware units and new elements representing software units.

15. The method according to claim 14, wherein in response to the bus traffic, isolation of the source code into the new elements representing hardware units and the new elements representing software units is optimized.

16. The method according to claim 11, further comprising:

upon determining that the source code is to be modified, modifying the source code by embedding the evaluation function one of immediately before or immediately after the line of source code in which the variables are written;

repeating the steps of sequentially reading in the source code, determining whether the source code is to be modified, and modifying at least one element of said source code elements until the source code is completely read in up to a last line of source code;

structuring the source code into elements representing the hardware units and elements representing the software units in the architecture design and compiling said source code;

organizing the compiled source code elements in a simulation program, executing the simulation program and calculating the data traffic on the bus;

calculating bus traffic using a known processing rate of the bus; and

performing evaluation of the performance of the bus in response to the bus traffic.

17. The method according to claim 16, wherein the variables loaded onto the bus consist of n bits while the bus consists of m bit lines, where n and m are both integers, and n is a multiple of m, and the bus traffic for the processing rate is produced such that the data traffic on the bus is multiplied by n/m and is then divided by the processing rate of the bus.

18. The method according to claim 11, wherein the general purpose high-level programming language is one of C language and C++ language.

19. The method according to claim 11, wherein the evaluation function increments a counting value if a pre-defined variable is loaded onto the bus.

20. A method in a LSI design and development process for facilitating and evaluating an architecture design for an algorithm design by performing a performance

evaluation of at least one bus at a high-level stage of said design and development process, said method comprises the steps of:

structuring a source code describing the algorithm design in a general purpose high-level programming language by isolating said source code into hardware elements and software elements interconnected by said bus having a bus configuration, creating the architecture design of an LSI;

creating an evaluation function for counting data traffic that occurs on said bus, wherein said evaluation function counts and represents a number of times the source code effects data transfer on said bus;

sequentially reading in said source code;

determining whether a line of source code represents writing data to variables that are defined in advance and are loaded onto the bus to be evaluated;

upon the determination, modifying the source code by embedding the evaluation function just before or after the line of source code in which one or more of the variables defined in advance are written;

repeating the steps of sequentially reading in the source code, determining whether the source code represents writing data to variables defined in advance, and modifying the source code by embedding the evaluation function until the source code is completely read in and modified up to a last line of source code;

compiling said structured source code elements;

organizing the compiled source code elements in a simulation program;

executing the simulation program and calculating the data traffic on the bus according to the evaluation function at the high level design stage;

calculating the bus traffic based on the data traffic and a known processing rate of the bus; and

performing the performance evaluation of the bus in response to the bus traffic being produced;

wherein said method is carried out again in response to the performance evaluation if said bus performance is not acceptable by starting with restructuring the source code and changing said bus configuration.

21. The method according to claim 20, further comprising:

feeding back the result of the performance evaluation of the bus to the step of structuring the source code to improve the architecture design at the high-level design stage by re-isolating said source code into hardware elements and software elements.

22. The method according to claim 20, wherein the bus configuration comprises the number of bit lines of the bus and, in response to the bus traffic, isolation of the source code into hardware elements and software elements is optimized by changing the number of bit lines of the bus.

23. The method according to claim 20, further comprising: after creating the evaluation function, sequentially reading in the source code line by line while effecting syntax analysis.

24. The method according to claim 20, wherein the variables loaded onto the bus consist of n bits while the bus consists of m bit lines, where n, m are both integral numbers, and n >= m, and the bus traffic for the processing rate is produced such that the data traffic on the bus is multiplied by n/m and is then divided by the processing rate of the bus.

25. The method according to claim 20, wherein the general purpose high-level programming language is C language or C++ language.

26. The method according to claim 20, wherein the evaluation function increments a counter value if a pre-defined variable is loaded onto the bus.